Port Knocking

flexible security through
authentication across
closed ports

Martin Krzywinski
Genome Sciences Centre

martink@bcgsc.ca www.bcgsc.ca

Port Knocking in 30 seconds

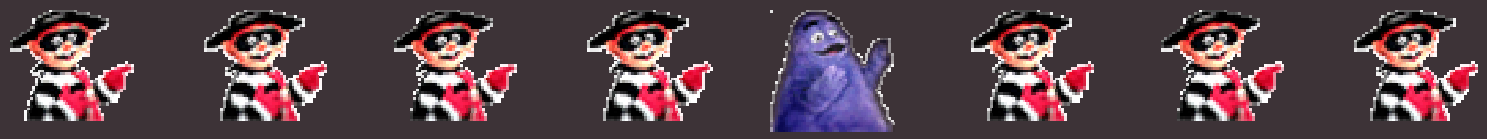
- method for granting access to hidden network services based on user identity checks
- identity check carried out by information transfer across closed ports
  - performed silently to viewpoint of user
  - mediated by connection attempts to encrypted, data-bearing port sequences (knocks)
- occludes network services from anyone failing silent identity checks
  - highly amenable to access control
  - illegitimate knocks are very loud and easily detected
- impossible to detect a port knocking server
  - cannot detect closed ports monitored by knocking daemon
- hard to intercept a port knocking transaction
  - authentication information travels one-way in a SYN packet
  - no actual data payload is sent
- early adopters benefit from the security by minority effect
  - *ceteris paribus*, if 1 person uses scheme A and 99 people use scheme B, breaking scheme B is more rewarding

Port Knocking in 3530 Seconds

- this holiday season, I want a security system that is
  - specific
    - all untrusted users are kept out
  - sensitive
    - all trusted users are let in
  - flexible
    - capable of variety of combinations of specificity and sensitivity
    - adapts to changing access requirements without impact on specificity and sensitivity
- and if I'm really good, let it also be
  - multi-layer and modular – defense in depth
  - robust and low impact
  - invisible, or at least subtle

Desirable Factor: Specificity

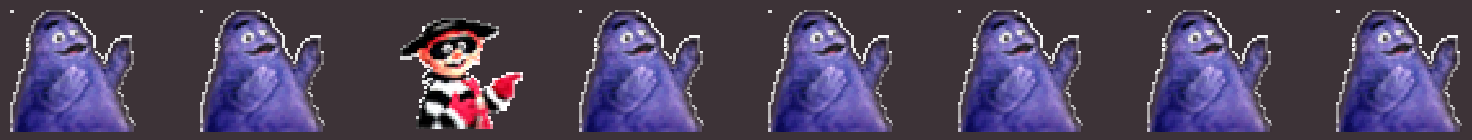
- security mechanisms categorize transactions
  - PASS or FAIL, or a derivative of this pair
  - similar to a statistical test
    - null hypothesis (assumption) = transaction is not allowed
    - apply packet/identity filters to reject assumption and PASS the transaction
- methods of categorization PASS/FAIL vary
  - packet filtering (IP), circuit level (TCP), application level
  - stateful multi-layer inspection (some combination of the above)
- any system must be extremely specific (FAIL when FAILABLE)
  - untrusted users (intruders) cannot be mistaken for trusted users
  - very small, preferably zero, false positive rate
  - false positives may result in a compromised system



a specific system detects all intruders

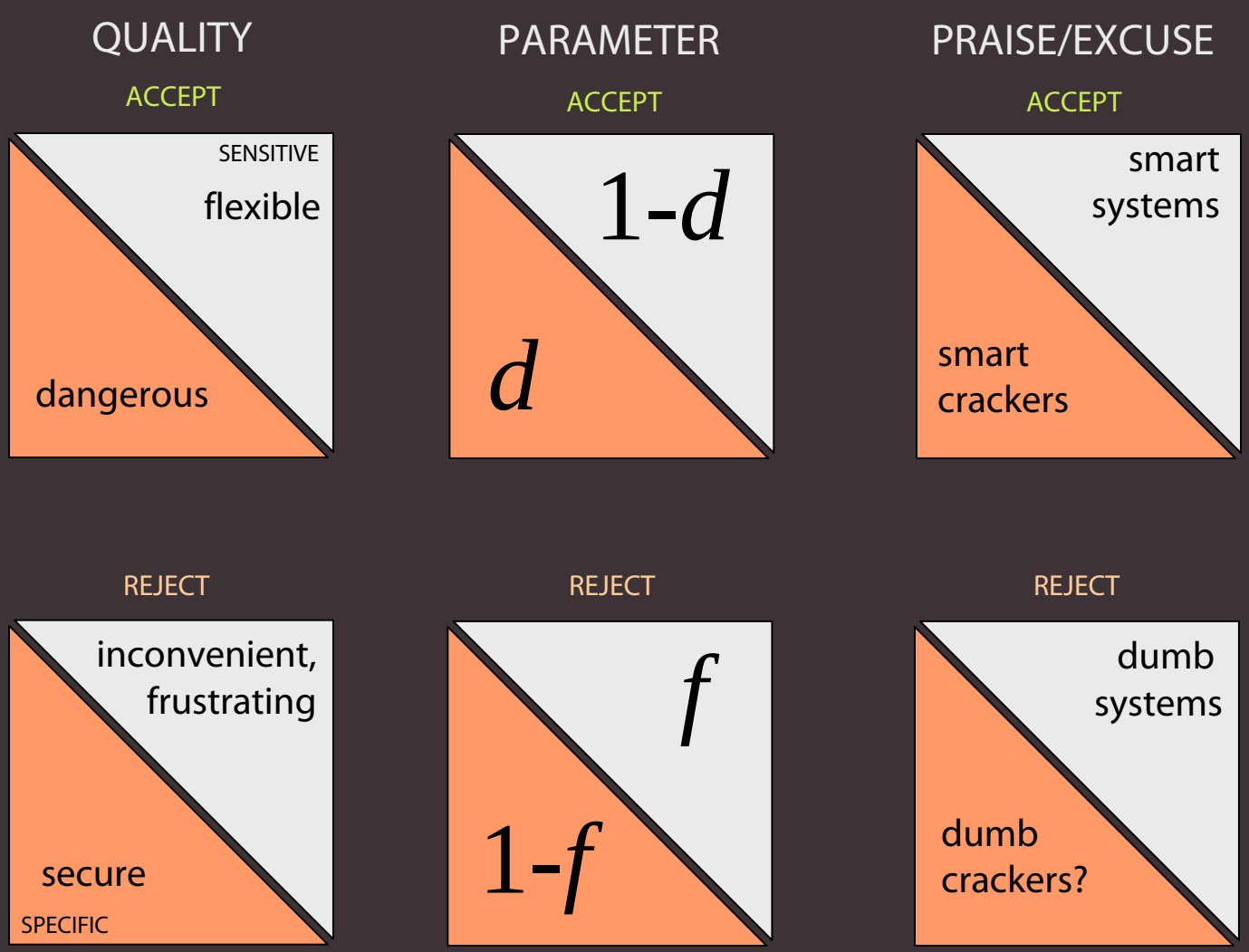
Desirable Factor: Sensitivity

- the system should be highly sensitive (PASS when PASSABLE)
  - discriminate trusted users from untrusted ones
  - small false negative rate
- lack of sensitivity produces false negatives
  - trusted users become frustrated
  - frustration drives opinions and policy
    - transfer to loss of confidence in specificity of system
    - relaxing security policies or abandoning the system
- a frustrated user is more acceptable than a compromised system
  - specificity trumps sensitivity



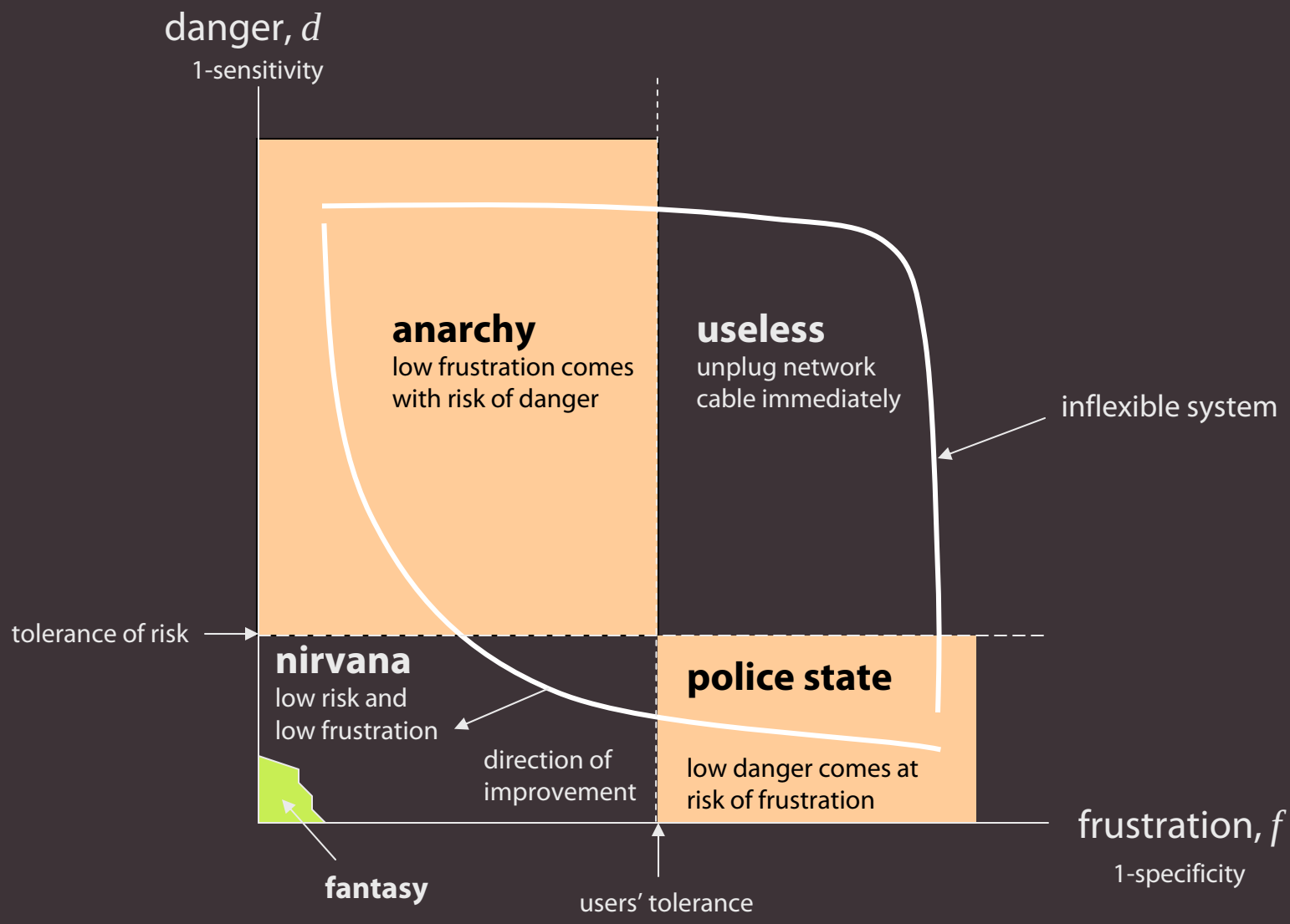
a sensitive system passes all trusted users

Quantifying Specificity and Sensitivity



Martin Krzywinski (martink@bcgsc.ca)

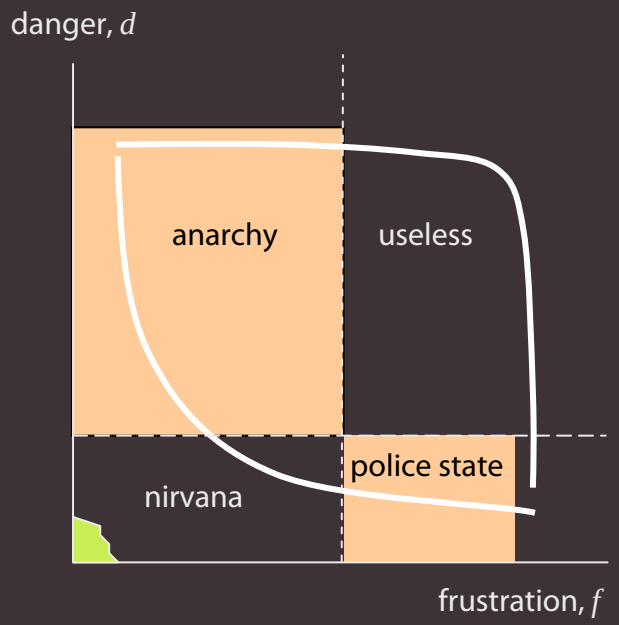
Danger vs Frustration – Decision Makers at Odds



Martin Krzywinski (martink@bcgsc.ca)

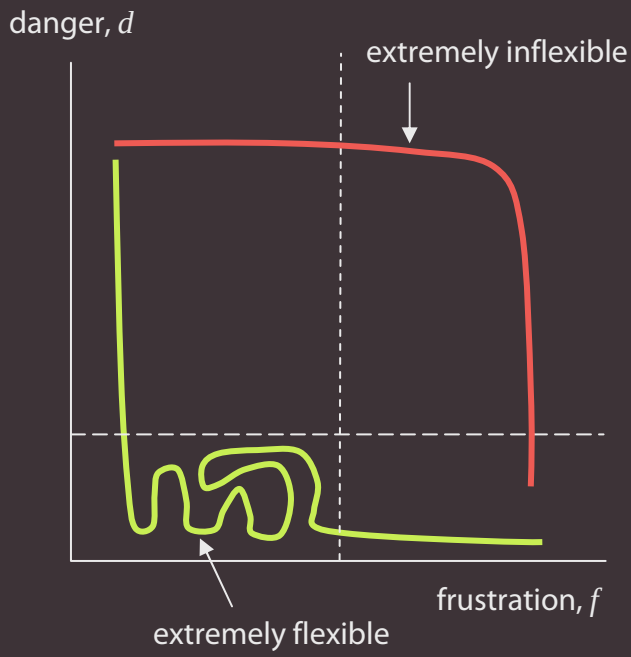
f and d Need to be Low

- $f \cdot d$ needs to be low
  - high f will lead to voluntary rejection of the system, even if d is low
  - high d will lead to forced rejection of system, even if f is low
- (most) people are smarter than (most) systems, given time
  - users circumvent frustration by finding gaps
  - …intruders circumvent safety (1-danger)
- trusted users expect systems to be smarter than they are
  - "why can't you know what I want?"
- trusted users fear that intruders are smarter than their systems
  - "how the hell did they get in?"



A Flexible System Samples Desirable (f ,d) Space

- when $f \cdot d$ is low, the system is flexible
  - adapts to changing behaviour of intruders and of trusted users
    - highly tunable parameters
- inflexible systems benefit from irreproducible factors
  - clairvoyant system administrators
  - magical properties of coincidence
- total flexibility is impossible to achieve because f, d are inter-related, competing, and do not compound geometrically
  - if either is zero, $f \cdot d$ is not zero
  - effective $f \cdot d + kd + k'f$, $k,k' > 0$
  - if both are zero, you're on a different planet
    - identity theft, social hacking, garbology
      - 9/10 surveyed at London's Waterloo station gave their passwords for a pen[§]
    - honest mistakes, dishonest mistakes



§ www.theregister.co.uk/content/55/30324.html

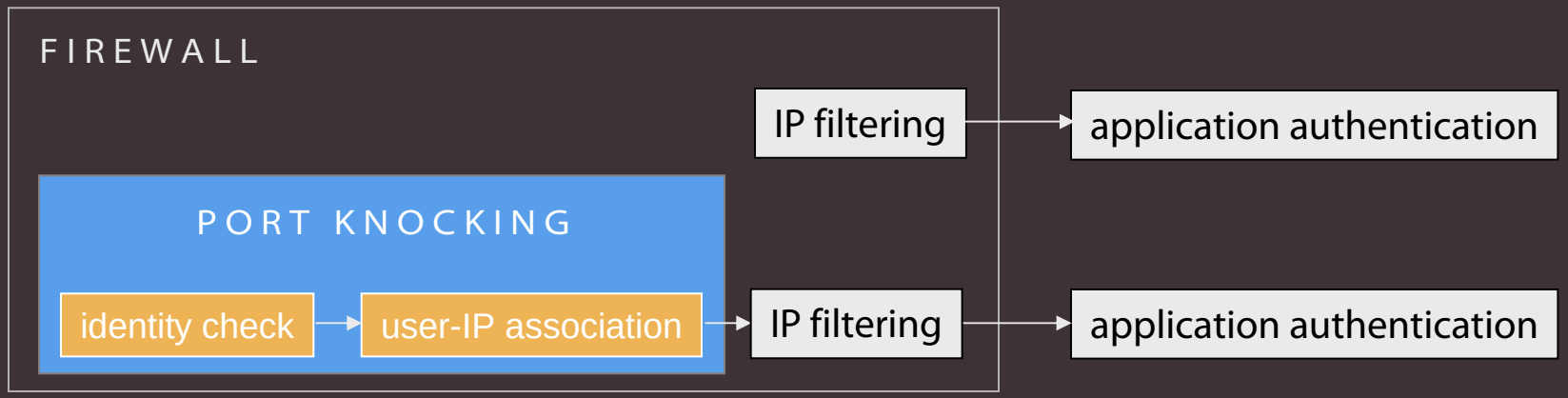
Martin Krzywinski (martink@bcgsc.ca)

(f ,d) with Packet Filtering and Application Security

- packet filtering firewalls and application security are common
  - hardware or software firewall
    - access rules based on remote/local IP and port
  - application security
    - personal security tokens (passwords, phrases, keys)
- firewall rules discriminate based on physical parameters of remote host
- application security relies on personal secret for identification
- firewall security predicated on well-documented, static canonical rule sets
  - changing host or port access lists may result in rules out of sync with requirements
  - static rule sets reduce flexibility, f
  - changing rule sets impact danger factor, d
- users and remote hosts do not obey 1:1 mapping
  - users change computers
  - increasing availability of access kiosks and cafes provide users with connectivity
  - maintaining static rules limits remote access

Need for Flexible Access Granting System

- biometric security tokens increasing in popularity
  - easy to ask someone for their password, harder for their biometric data
    - I don't know my fingerprint the way I know my password
  - consider phones – I can use any phone to call my friend Bob because Bob can identify me
  - consider computers – I cannot user any computer because my firewall cannot identify me
    - why should I care that I'm using a different computer
- filtering by IP limits individual access
  - IP filtering suitable between immobile elements
    - organizations, groups, processes
  - IP filtering unsuitable when one of the communication nodes is highly mobile
    - travel, collaboration

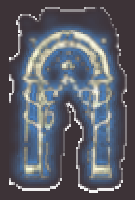


Firewall for Identity Checking – Port Knocking

- TCP connection attempts initiated by remote users act as an identity check
  - firewall becomes the authenticating application
  - closed ports are the "keyboard keys" for "typing" the password
- lowers frustration factor, *f*, because trusted users are no longer limited to trusted IPs
- lowers danger factor, *d*, because network services (even hosts) are invisible
  - permits networked resources to be hidden and undetectable unless user identity is verified
- why hide resources?



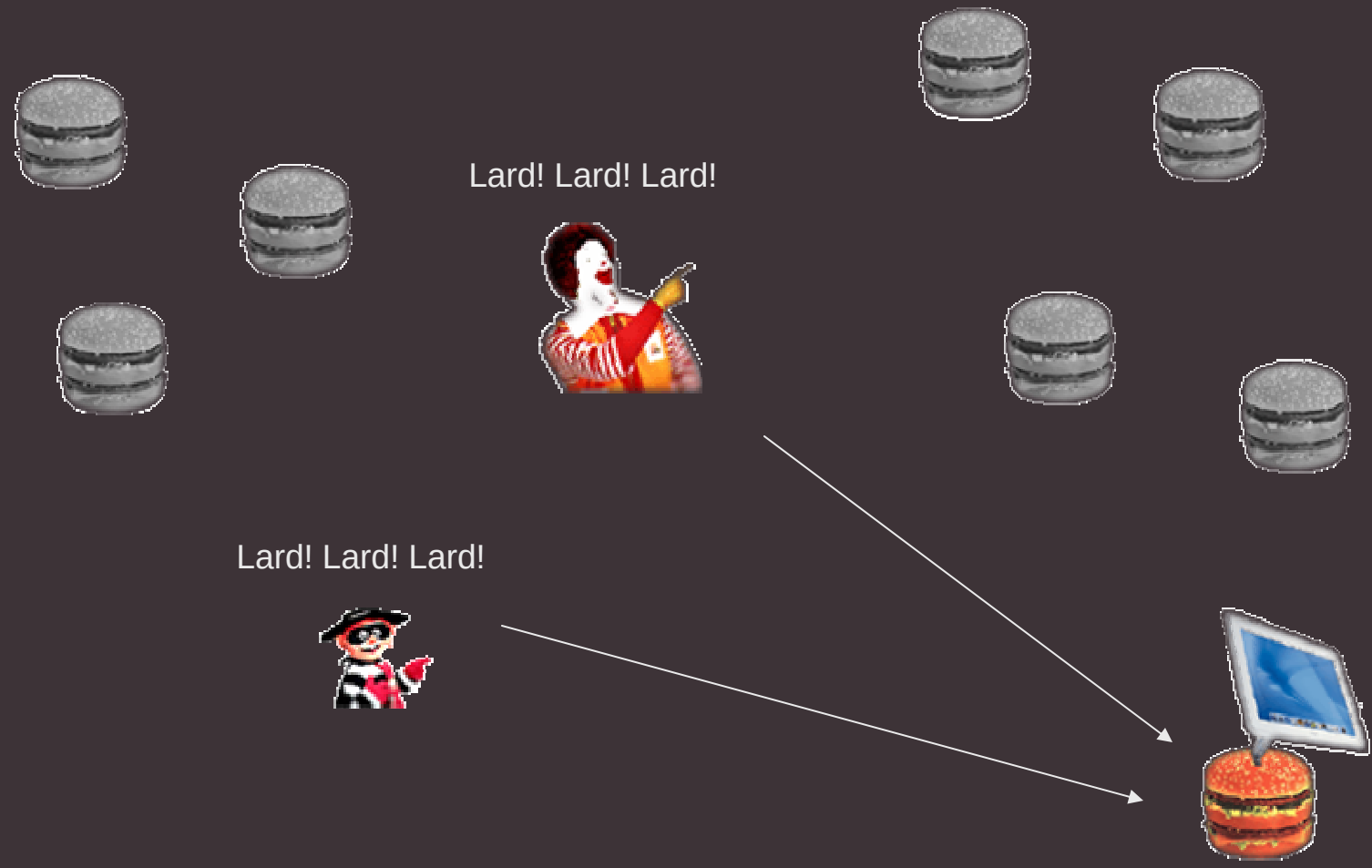
trippicket 1.1

securhund 0.5

durindoor 2.1


> telnet xx.xx.xx.xx yy
trying xx.xx.xx.xx…
connected to securehost.securisnazz.com
Escape character is '^]'
running trippicket 1.1, securhund 0.2, durindoor 0.1

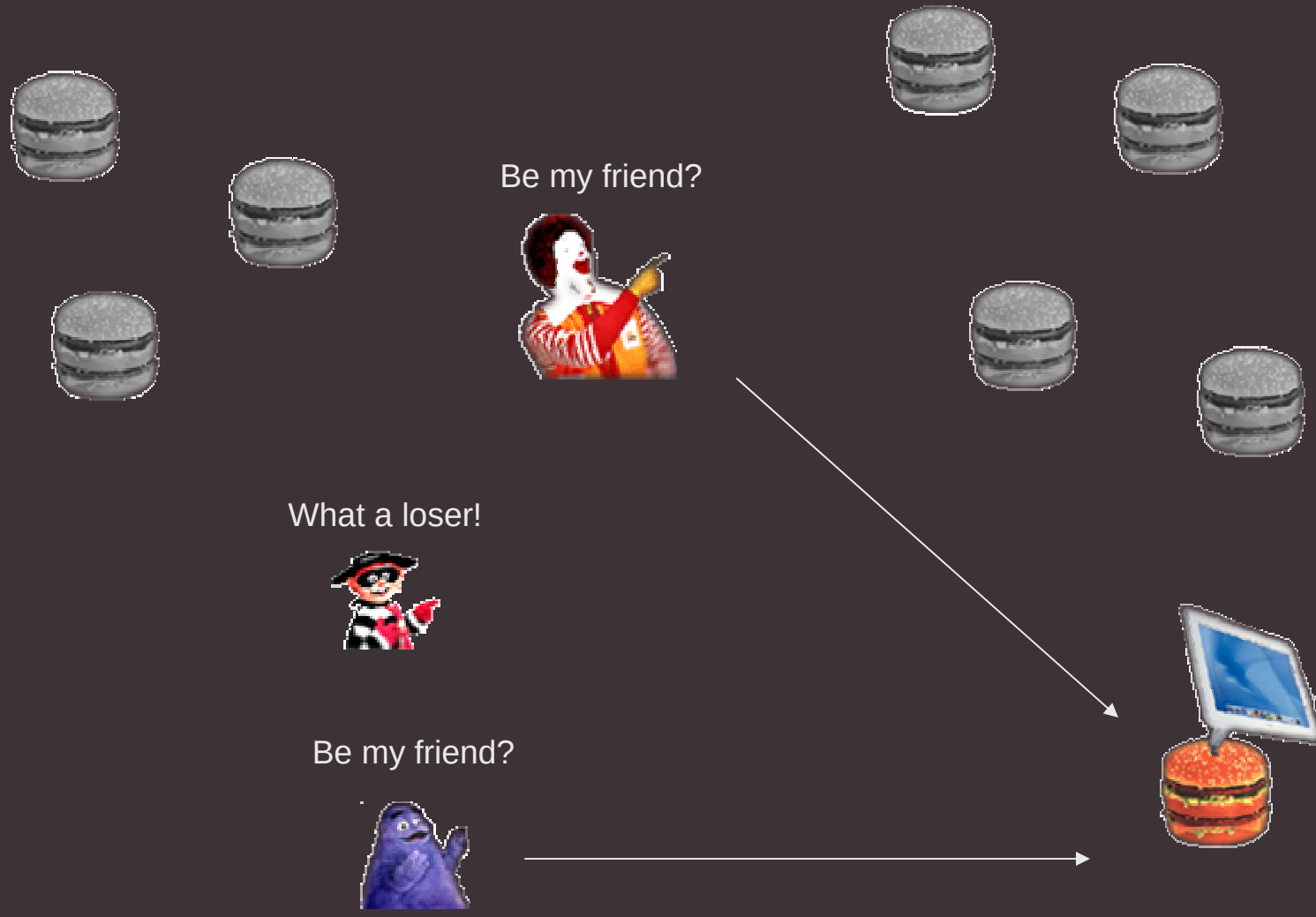
Login:
Password rejected! We are secure!

Invisible Triggering Processes: Hide Service Not Security



Martin Krzywinski (martink@bcgsc.ca)

Non-Intuitive Triggers



Martin Krzywinski (martink@bcgsc.ca)

Personal Encrypted Triggers

name ron
vision wavelength 556.3nm
appetite bigmac

password iatebillions

encrypted+encoded trigger 4af2 8d2e 820b 82cc a37d 002a

name h.b.
vision wavelength 553.3nm
appetite bigmac

guess password
decrypt
substitute
encrypt

45f2 26ff bd3a 78b2 aa32 7cf21

(vision 553.3 nm)

Trigger Service is the Outer Defense Layer

- encrypt public information with private secret to reveal hidden available resources
  - additional security measures are still in place
  - invisible trigger services provide means to hide your resources

4af2 8d2e 820b
82cc a37d 002aB



- trigger detector is independent of all other security and authentication services

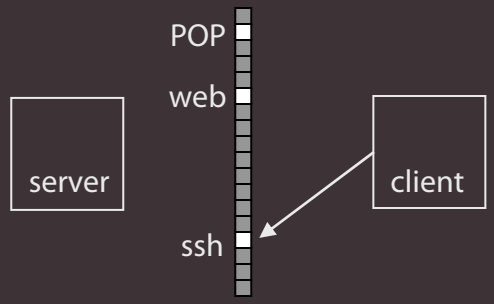
- is this obscurity?[§]
  - not as long as good access control is maintained
    - know who's doing what, to whom, how and when
  - cryptographically strong encryption
    - keep algorithms public and personal information private
  - force attackers to be less stealthy
    - why is h.b. yelling random phrases with a hungry look in his eye in an otherwise quiet room?
    - hiding in an empty room makes it easier to detect attackers

[§] www.bastille-linux.org/jay/obscurity-revisited.html

Port Knocking in Practice

Open application policy

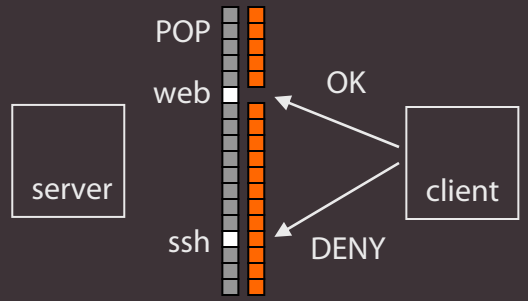
server running ssh, web and POP



client can detect ssh, web, POP service
client can attempt to authenticate with all services
client can try to break into all services

Firewalled applications

server running a firewall blocking ssh from client



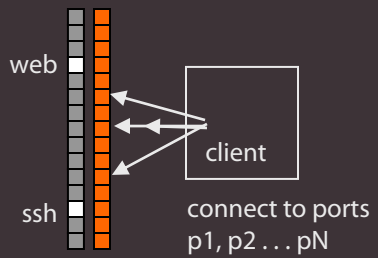
client cannot detect that ssh is running
client cannot detect that POP is not running
client cannot authenticate with ssh service
client cannot break into ssh application

Port Knocking in Practice

STEP 1

Knocking Phase

client knocks on
N closed ports



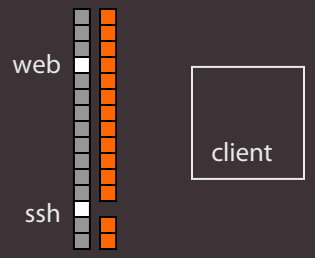
no data sent back to client

client *a priori* cannot tell whether knocking daemon is listening

STEP 2

Firewall Rule Relaxation

server responds to
authentic knock



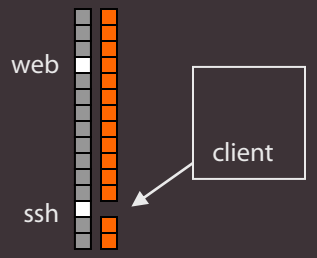
daemon opens ssh port to client IP for 30 minutes

response to knock completely arbitrary (e.g. disallow second identical port knock attempt)

STEP 3

Client Starts Session

client connects and
authenticates with application



client connects to ssh and authenticates with system password

Step 1 – The Knock

- the knock is an integer-encoded encrypted string which may contain information such as
  - client's IP
  - requested port or range of ports to open
  - expected session time
  - additional parameter flags or commands
- encryption of knock should be strong
  - one-time pads for connection from highly untrusted locations

142 103 205 1 22 15 233 → 572 500 742 721 526 637 741 609

no IV, Blowfish, "password"

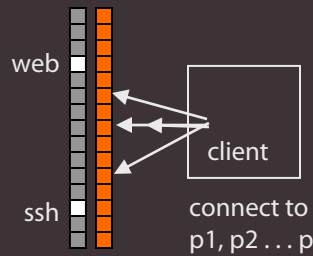
142 103 205 1 22 15 233 → 582 597 610 600 611 609 573 586
573 606 600 610 730 516 744 731
632 710 681 748 637 537 573 628
605 574 659 574 677 557 711 682

IV, Twofish, "vcwpnepflozkxbfrzydf"

STEP 1

Knocking Phase

client knocks on
N closed ports



connect to ports
p1, p2 . . . pN

The Knock is Mediated by Firewall Log File

- knocks are transmitted as connection attempts
- client does not receive ICMP error packets

CLIENT

```
> telnet FIREWALL 102
> telnet FIREWALL 100
> telnet FIREWALL 100
> telnet FIREWALL 103
```

SERVER

```
> tail –f firwewall.log
Feb 12 00:13:26 ... input DENY ... CLIENT:64137 FIREWALL:102 ...
Feb 12 00:13:27 ... input DENY ... CLIENT:64138 FIREWALL:100 ...
Feb 12 00:13:27 ... input DENY ... CLIENT:64139 FIREWALL:100 ...
Feb 12 00:13:28 ... input DENY ... CLIENT:64140 FIREWALL:103 ...
```

- information is sent across closed ports
  - information content limited by knock length and encoding
- a listening knocking server is undetectable by direct probing
- illegitimate knocks are very loud
  - flexible access control

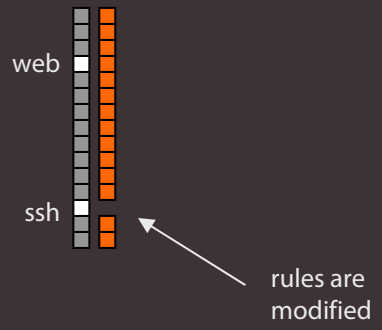
Step 2 – Knock Daemon Response

- the knock must contain client's IP
  - client can act as a knocking proxy and use a 3rd party IP address
- knock daemon maintains a queue of all connection attempts to predetermined range of ports
  - errors in knocks due to routing hard, not impossible, to fix
  - knocks may contain checksums and redundant payload
- daemon response to knock is arbitrary
  - modify firewall rules
    - open/close a port
  - deny further connection attempts
  - shut down, send mail, do backups
- knock daemon reveals resources to the client
- post-knock IP filtering
  - other firewall rules can apply

STEP 2

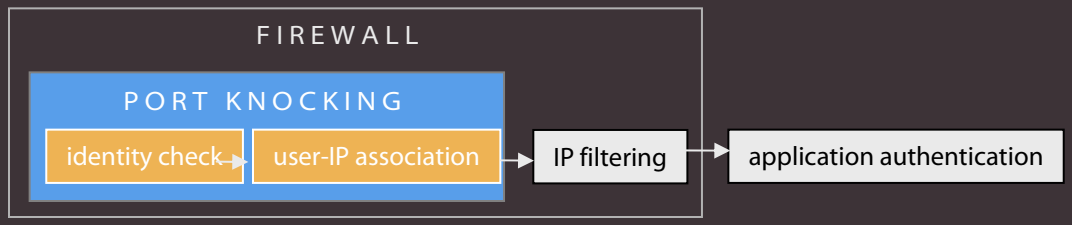
Firewall Rule Relaxation

server responds to authentic knock



Step 3 – Initiating the Session

- client connects as usual

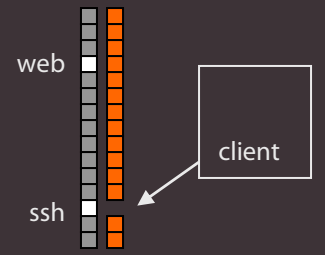


- knock may contain paranoia safeguards
  - request that daemon does not acknowledge additional knocks from client
  - request that daemon refuse additional connections from client

STEP 3

Client Starts Session

client connects and authenticates with application



Benefits of Port Knocking

- prospect of maintaining very sensitive data nearline – offline but accessible
  - periodic monitoring via ssh of remote server
  - hidden frontdoors for service personnel
  - manually initiated processes using port knocking triggers
- occluding resources limits their exposure to exploit attempts
  - still patch regularly, but no need to rush back from vacation
- independent authentication system using firewall
  - robust
  - independent of OS if firewall IP stack independent
  - use of intrusion detection systems (IDS) augments knock daemon's ability to spot scans, knock hunts, illegitimate knocks
- transition from IP/user-centric to pure user-centric authentication
- obviates need to alter firewall rules to follow traveling users
- frustration and danger reduced

Potential Disadvantages

- conscious use of knock client required
  - novel implementations may accept subconscious use
- preserving knock integrity difficult in congested environments
  - ordinality of packets not necessarily preserved
  - develop knocks resistant to shuffling
- complex knock queue for multiple clients behind remote gateways
  - multiple users hiding behind single IP
  - users can initiate on-demand-access to remote services
    - can become very complex

Knocking on Blue Sky

- hardware implementation
  - corporate, business, SOHO, home devices
  - home routers already have port forwarding and triggering
- autonomous, rechargeable clients on portable media
  - biometric USB key performs knock using fingerprint
  - users cannot give away the knock for a pen
- alternative forms of authentication will be required
  - user population increasingly more mobile
  - connections from unpredictable locations
  - associating users with specific computers or networks will cease to be practical

References and Acknowledgements

- I would like to thank
  - Hardondel Sibble
    - www.pdscc.com
  - Mark Mayo
    - Genome Sciences Centre Information Systems Coordinator
    - www.permeta.com
  - Ian Bosdet, Duane Smailus

- Port Knocking publications
  - Linux Journal, June 2003
    - www.linuxjournal.com/article.php?sid=6811
  - SysAdmin Magazine, June 2003
    - www.samag.com/articles/2003/0306/

- WCSF 2003 organizers and Board

Martin Krzywinski (martink@bcgsc.ca)

Port Knocking

flexible security through
authentication across
closed ports

Martin Krzywinski
Genome Sciences Centre

martink@bcgsc.ca www.bcgsc.ca